# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-006815
(43)Date of publication of application : 12.01.1996

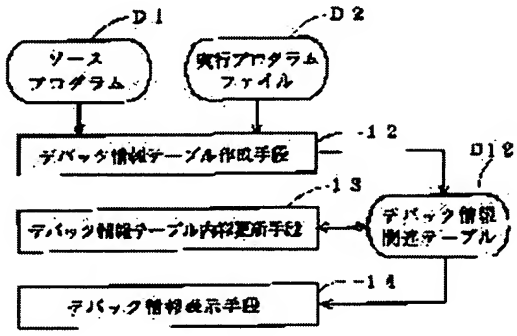
(51)Int.CI. G06F 11/28
G06F 9/06
G06F 9/45

(21)Application number : 06-139191
(22)Date of filing : 21.06.1994

(71)Applicant : MITSUBISHI ELECTRIC CORP
(72)Inventor : TSUBOTA HIRONO
TAMURA TOSHIYUKI
KOMORI NOBUFUMI

## (54) DEBUGGING DEVICE, COMPILER, AND STATISTICAL ANALYZING DEVICE

(57)Abstract:
PURPOSE: To provide a debugging device which can grasp the accurate execution state of, specially, a parallel processing system.
CONSTITUTION: A debugging information relative table generating means 12 generates a debugging information relative table group D12 on the basis of a source program D1 and an execution program D2 (including debugging information). A debugging information relative table contents updating means 13 receives program execution information including variable update execution information from the parallel processing system when an instruction corresponding to variable update on the source program D1 is executed by the parallel processing system, and updates the contents of the debugging information relative table group D12 on the basis of the variable update execution information in the program execution information. A debugging information display means 14 displays debugging display information which is easy to visually recognize on the basis of the debugging information relative table group D12.




LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision

of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

* NOTICES *

JPO and NCIPI are not responsible for any
damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

---

## CLAIMS

---

[Claim(s)]
[Claim 1] It is debugging equipment which debugs the computing system in which the output of the program execution information which operates based on an executive program and shows the program execution situation is possible. A source program grant means to give the source program which consists of two or more instruction statement which operates said computing system, It has an executive program generation means to generate said executive program based on said source program. Said executive program includes the information which shows relation with said source program. It is based on said source program, said executive program, and said program execution information. A renewal means of computing system performance information to always update said computing system performance information according to actuation of said computing system which generates the computing system performance information which shows the situation of said computing system of operation, and changes every moment, It has further a display means to display the computing system display information which can be recognized visual based on said computing system performance information. Said computing system performance information The information which identifies the existing run statement to which activation was already performed among said two or more instruction statement of said source program, and the non-run statement which is not yet performed is included. Said display means Debugging equipment characterized by displaying said computing system display information that said existing run statement and non-run statement in said source program can identify visually.
[Claim 2] It is debugging equipment which debugs the computing system in which the output of the program execution information which operates based on an executive program and shows the program execution situation is possible. A source program grant means to give the source program which consists of two or more instruction statement which operates said computing system, It has an executive program generation means to generate said executive program based on said source program. Said executive program includes the information which shows relation with said source program. It is based on said source program, said executive program, and said program execution information. A renewal means of computing system performance information to always update said computing system performance information according to actuation of said computing system which generates the computing system performance information which shows the situation of said computing system of operation, and changes every moment, It has further a display means to display the computing system display information which can be recognized visual based on said computing system performance information. Said computing system performance information The information which identifies the count of activation of each of two or more of said instruction statement of said source program is included. Said display means Debugging equipment characterized by the count of activation displaying that said computing system display information is visually discriminable to said each of two or more instruction statement in said source program.
[Claim 3] It is debugging equipment which debugs the computing system in which the output of the program execution information which operates based on an executive program and shows the program execution situation is possible. A source program grant means to give the source program which consists of two or more instruction statement which operates said computing

system, It has an executive program generation means to generate said executive program based on said source program. Said executive program includes the information which shows relation with said source program. It is based on said source program, said executive program, and said program execution information. A renewal means of computing system performance information to always update said computing system performance information according to actuation of said computing system which generates the computing system performance information which shows the situation of said computing system of operation, and changes every moment, It has further a display means to display the computing system display information which can be recognized visual based on said computing system performance information. Said computing system performance information The information which shows the activation hysteresis of each instruction statement within a loop formation which is the instruction statement which forms a loop formation among said two or more instruction statement is included. Said display means Debugging equipment characterized by the activation hysteresis displaying that said computing system display information is visually discriminable to each instruction statement within said loop formation.

[Claim 4] It is debugging equipment which debugs the computing system in which the output of the program execution information which operates based on an executive program and shows the program execution situation is possible. A source program grant means to give the source program which consists of two or more instruction statement which operates said computing system, It has an executive program generation means to generate said executive program based on said source program. Said executive program includes the information which shows relation with said source program. It is based on said source program, said executive program, and said program execution information. A renewal means of computing system performance information to always update said computing system performance information according to actuation of said computing system which generates the computing system performance information which shows the situation of said computing system of operation, and changes every moment, It has further a display means to display the computing system display information which can be recognized visual based on said computing system performance information. Said two or more instruction statement contains the share function instruction statement which constitutes a share function. Said computing system performance information Said display means is debugging equipment characterized by displaying said computing system display information that said identifier can identify visually corresponding to said share function instruction statement including the information about the identifier for identifying share function call origin.

[Claim 5] Said share function is debugging equipment according to claim 4 which is a recursive function.

[Claim 6] It is debugging equipment which debugs the computing system in which the output of the program execution information which operates based on an executive program and shows the program execution situation is possible. A source program grant means to give the source program which consists of two or more instruction statement which operates said computing system, It has an executive program generation means to generate said executive program based on said source program. Said executive program includes the information which shows relation with said source program. It is based on said source program, said executive program, and said program execution information. A renewal means of computing system performance information to always update said computing system performance information according to actuation of said computing system which generates the computing system performance information which shows the situation of said computing system of operation, and changes every moment, It has further a display means to display the computing system display information which can be recognized visual based on said computing system performance information. Said two or more instruction statement contains the function instruction statement which constitutes two or more functions. Said computing system performance information Said display means is debugging equipment characterized by the count of activation just before said two or more functions displaying that said computing system display information is visually discriminable including the information which identifies the count of activation just before being a count of activation in a past fixed period of a just before [ said two or more functions ].

[Claim 7] A source program reading means to be the compiler which generates the executive program for predetermined computing systems, and to read the source program which consists of two or more instruction statement which operates said computing system, It has a program code group generation means to generate the program code group of an executive program format based on said source program. As opposed to specific instruction statement [ in / in said program code group generation means / said two or more instruction statement ] An optimization processing means to generate a different program code for every generation condition of the, to receive said program code group, to perform optimization processing to said program code group, and to output an optimization processing finishing program code group, The compiler further equipped with a program code conversion means to change all the program codes corresponding to said specific instruction in said optimization processing finishing program code group into the same code, and to output said executive program.
[Claim 8] Said predetermined computing system is a compiler according to claim 7 said whose specific instruction parallel computing processing is possible and is a synchronizing primitive during two or more processings.
[Claim 9] It is statistical analysis equipment which performs statistical analysis of the contents of processing of the computing system which operates based on the executive program which consists of two or more instructions. An information reading means for statistics to read the information for statistics including the information which can detect the activation hysteresis of said computing system at least, A statistical analysis command grant means to give the statistical analysis command information that the instruction combination used as a statistical analysis object is directed, Statistical analysis equipment equipped with a statistical analysis means to receive said information for statistics, and said statistical analysis command information, to count the count of activation of the instruction combination which said statistical analysis command information directs with reference to said information for statistics, to perform statistical analysis, and to output a statistical analysis result.
[Claim 10] Said statistical analysis command information is statistical analysis equipment according to claim 9 which is the information which directs the instruction combination which can be performed to juxtaposition.
[Claim 11] Said statistical analysis command information is statistical analysis equipment according to claim 9 which is the information which directs instruction combination with a data dependency.
[Claim 12] Said statistical analysis means is statistical analysis equipment according to claim 11 which also performs statistical analysis about the execution time of the instruction combination which has said data dependency further.
[Claim 13] It is statistical analysis equipment given in any 1 term of claim 9 to which said executive program consists of two or more partial programs, and said statistical analysis means outputs the statistical analysis result in said two or more partial-program units thru/or claim 12.

---

[Translation done.]

* NOTICES *

JPO and NCIPI are not responsible for any
damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

## DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Industrial Application] This invention relates to the statistical analysis equipment which performs statistical analysis of the contents of processing of the computing system which operates based on an executive program and operates based on the executive program which consists of the debugging equipment which debugs the computing system in which the output of the program execution information which shows that program execution situation is possible, a compiler which generates the executive program for predetermined computing systems, and two or more instructions.
[0002]
[Description of the Prior Art] Although various development exchange equipments are also developed in order to develop many parallel processing systems and to support development of this parallel processing system in recent years, it is a still unripe phase.
[0003] In the conventional system, since processing was carried out to the sequence described by the source program, the display of the activation situation of a symbolic debugger displayed only the part where current activation is performed with the run designation of an arrow head etc., and has grasped the activation situation as indicated by 209-210 pages of "C development environment for user ZUGAIDO THINK C Macintosh" of SYMANTEC.
[0004] However, with the parallel processing system, since two or more processings were performed to juxtaposition, it might be said like the conventional symbolic debugger that exact grasp of an activation situation could not be performed only by displaying a current activation part.
[0005] In a parallel processing system, since two or more processings are performed by juxtaposition, in order to avoid the side effect of memory access, synchronous processing is needed for the schedule of a task etc. Therefore, the code of synchronous processing is generated by various conditions in the linguage processing system (compiler) of a parallel processing system.
[0006] Since the code of synchronous processing is locally generated for every various conditions, it becomes redundancy in many cases. Moreover, it may become unnecessary with combination with other optimization. For this reason, the demand of optimization is large. Moreover, in order to perform more advanced optimization of synchronous processing, it is necessary to carry out in a different phase after all synchronous processing code generation.
[0007] Since synchronous processing was conventionally realized by one synchronizing primitive sync, it was difficult to perform fine optimization for every conditions.
[0008] That is, about the linguage processing systems, such as a compiler, with a parallel processing system, since two or more processings were performed to coincidence, the sequentiality of processing was guaranteed, and the code for synchronous processing needed to be added, there is a trouble that the processing engine performance falls by this synchronization, and an efficient executive program was not able to be generated. in addition -- training of optimization of a synchronization -- Information Processing Society of Japan 48th -- it is described by time national conference lecture collected-works (6)4B-10 page 6-65.

[0009] Furthermore, especially a parallel processing system is used for the application with which a rate is demanded in many cases. In developing highly efficient CPU, there was a demand of wanting to make one instruction instruction combination performed frequently, and to accelerate with the application, but when making which instruction combination one instruction, there was a trouble that it was difficult to grasp whether it is efficient.
[0010]
[Problem(s) to be Solved by the Invention] There was a trouble that it could not grasp whether it is efficient if the development support system of the conventional parallel processing system is constituted as mentioned above, it cannot generate the efficient executive program which grasp of the exact activation situation of a parallel processing system is [ executive program ] difficult, and operates a parallel processing system in the conventional compiler but which instruction combination is made one instruction in a parallel processing system with conventional debugging equipment.
[0011] It was made in order that this invention might solve the above-mentioned trouble, and if which instruction combination is made one instruction especially in a parallel processing system, it will aim at obtaining the statistical analysis equipment which can grasp [ the linguage processing system which generates the debugging equipment which can grasp the exact activation situation of especially a parallel processing system, especially the efficient executive program which operates a parallel processing system, and ] whether it is efficient.
[0012]
[Means for Solving the Problem] The debugging equipment according to claim 1 concerning this invention It is equipment which debugs the computing system in which the output of the program execution information which operates based on an executive program and shows the program execution situation is possible. A source program grant means to give the source program which consists of two or more instruction statement which operates said computing system, It has an executive program generation means to generate said executive program based on said source program. Said executive program includes the information which shows relation with said source program. It is based on said source program, said executive program, and said program execution information. A renewal means of computing system performance information to always update said computing system performance information according to actuation of said computing system which generates the computing system performance information which shows the situation of said computing system of operation, and changes every moment, It has further a display means to display the computing system display information which can be recognized visual based on said computing system performance information. Said computing system performance information The information which identifies the existing run statement to which activation was already performed among said two or more instruction statement of said source program, and the non-run statement which is not yet performed is included. Said display means Said computing system display information is displayed that said existing run statement and non-run statement in said source program can identify visually.
[0013] Moreover, in debugging equipment according to claim 2, said computing system performance information displays said computing system display information that the count of activation can identify said display means visually to said each of two or more instruction statement in said source program including the information which identifies the count of activation of each of two or more of said instruction statement of said source program.
[0014] Moreover, in debugging equipment according to claim 3, said computing system display information is displayed that the activation hysteresis can identify said display means visually to each instruction statement within said loop formation including the information which shows the activation hysteresis of each instruction statement within a loop formation which is the instruction statement in which said computing system performance information forms a loop formation among said two or more instruction statement.
[0015] Moreover, in debugging equipment according to claim 4, said computing system display information is displayed that said identifier can identify said display means visually corresponding to said share function instruction statement including the information about an identifier for said computing system performance information to identify share function call origin including the

share function instruction statement from which said two or more instruction statement constitutes a share function.
[0016] Furthermore, like debugging equipment according to claim 5, said share function may be constituted so that it may be a recursive function.
[0017] Moreover, in debugging equipment according to claim 6, said computing system display information is displayed that the count of activation just before said two or more functions can identify said display means visually including the information which identifies the count of activation just before said computing system performance information is a count of activation in a past fixed period of a just before [ said two or more functions ] including the function instruction statement from which said two or more instruction statement constitutes two or more functions.
[0018] The compiler according to claim 7 concerning this invention A source program reading means to read the source program which consists of two or more instruction statement which the executive program for predetermined computing systems is generated [ instruction statement ], and operates said computing system, It has a program code group generation means to generate the program code group of an executive program format based on said source program. As opposed to specific instruction statement [ in / in said program code group generation means / said two or more instruction statement ] An optimization processing means to generate a different program code for every generation condition of the, to receive said program code group, to perform optimization processing to said program code group, and to output an optimization processing finishing program code group, It has further a program code conversion means to change all the program codes corresponding to said specific instruction in said optimization processing finishing program code group into the same code, and to output said executive program, and is constituted.
[0019] Moreover, like a compiler according to claim 8, parallel computing processing is possible for said predetermined computing system, and it may constitute said specific instruction so that it may be a synchronizing primitive during two or more processings.
[0020] The statistical analysis equipment according to claim 9 concerning this invention It is equipment which performs statistical analysis of the contents of processing of the computing system which operates based on the executive program which consists of two or more instructions. An information reading means for statistics to read the information for statistics including the information which can detect the activation hysteresis of said computing system at least, A statistical analysis command grant means to give the statistical analysis command information that the instruction combination used as a statistical analysis object is directed, Said information for statistics and said statistical analysis command information are received, and it has a statistical analysis means to count the count of activation of the instruction combination which said statistical analysis command information directs with reference to said information for statistics, to perform statistical analysis, and to output a statistical analysis result, and is constituted.
[0021] Moreover, said statistical analysis command information may be information which directs the instruction combination which can be performed to juxtaposition like statistical analysis equipment according to claim 10.
[0022] Moreover, said statistical analysis command information may be information which directs instruction combination with a data dependency like statistical analysis equipment according to claim 11.
[0023] In addition, like statistical analysis equipment according to claim 12, configuration ****** of said statistical analysis means is also good so that statistical analysis about the execution time of the instruction combination which has said data dependency further may also be performed.
[0024] Moreover, like statistical analysis equipment according to claim 13, said executive program consists of two or more partial programs, and said statistical analysis means may be constituted so that the statistical analysis result in said two or more partial-program units may be outputted.
[0025]

[Function] In the debugging equipment according to claim 1 in this invention computing system performance information The information which identifies the existing run statement to which activation was already performed among two or more instruction statement of a source program, and the non-run statement which is not yet performed is included. Since computing system display information is displayed that the existing run statement and the non-run statement in a source program can identify a display means visually, the existing run statement and the non-run statement in a source program can be easily recognized only by referring to computing system display information.

[0026] It sets to the debugging equipment according to claim 2 in this invention, and since the count of activation displays that computing system display information is visually discriminable to two or more instruction statement of each which can set a display means to a source program including the information from which computing system performance information discriminates the count of activation of two or more instruction statement of each of a source program, the count of two or more instruction statement of each, which can be easily set to a source program only by referring to computing system display information, of activation can recognize.

[0027] In the debugging equipment according to claim 3 in this invention computing system performance information A display means receives each instruction statement within a loop formation including the information which shows the activation hysteresis of each instruction statement within a loop formation which is the instruction statement which forms a loop formation among two or more instruction statement. Since the activation hysteresis displays that computing system display information is visually discriminable, activation situations, such as loop count of the instruction statement within a loop formation in a source program, can be easily recognized only by referring to computing system display information.

[0028] In the debugging equipment according to claim 4 in this invention, since computing system display information is displayed that an identifier can identify a display means visually corresponding to share function instruction statement including the information about an identifier for computing system performance information to identify share function call origin, the identifier of the share function instruction statement in a source program can be easily recognized only by referring to computing system display information.

[0029] Moreover, since the above-mentioned share function displays computing system display information that the account identifier of the high recursive function of possibility of being called frequently can identify visually, debugging equipment according to claim 5 can recognize the identifier of the recursive function instruction statement in a source program easily only by referring to computing system display information.

[0030] In the debugging equipment according to claim 6 in this invention computing system performance information In order to display computing system display information that the count of activation just before two or more functions can identify a display means visually including the information which identifies the count of activation just before being a count of activation in a past fixed period of a just before [ two or more functions ], The count of activation just before two or more functions which can be easily set to a source program only by referring to computing system display information can be recognized.

[0031] Since the program code group generation means of the compiler according to claim 7 concerning this invention generates a different program code for every generation condition of that to the specific instruction statement in two or more instruction statement, an optimization processing means can perform the part which can perform optimization processing in consideration of the generation condition of specific instruction statement, and more advanced optimization processing.

[0032] Since the above-mentioned specific instruction is a synchronizing primitive, especially a compiler according to claim 8 can generate an efficient executive program for a predetermined computing system to perform a parallel processing system.

[0033] The statistical-analysis means in the statistical-analysis equipment according to claim 9 concerning this invention receives the information for statistics, and statistical-analysis command information, and since it counts the count of activation of the instruction combination which statistical-analysis command information directs with reference to the information for

statistics, performs statistical analysis and outputs a statistical-analysis result, it can recognize the activation frequency of the instruction combination concerned by referring to a statistical-analysis result.
[0034] Moreover, since the statistical analysis command information on statistical analysis equipment according to claim 10 is information which directs the instruction combination which can be performed to juxtaposition, it can recognize the activation frequency of the instruction combination which can be performed to juxtaposition by referring to a statistical analysis result.
[0035] Moreover, since the statistical analysis command information on statistical analysis equipment according to claim 11 is information which directs instruction combination with a data dependency, it can recognize the activation frequency of instruction combination with a data dependency by referring to a statistical analysis result.
[0036] Moreover, since the statistical analysis means of statistical analysis equipment according to claim 12 also performs statistical analysis about the execution time of the instruction combination which has a data dependency further, it can also acquire the information about the execution time of the instruction combination concerned by referring to a statistical analysis result.
[0037] Moreover, since a statistical analysis means according to claim 13 outputs the statistical analysis result in two or more partial-program units, it can grasp which instruction combination it is adapted for every partial program, and should be made one instruction.
[0038]
[Example]
< <development support system whole configuration >> drawing 1 is the explanatory view having shown the development support system of a parallel processing system typically. As shown in this drawing, the development support system of a parallel processing system 4 consists of a symbolic debugger 1, a compiler 2, and instruction combination statistical analysis equipment 3.
[0039] A compiler 2 reads a source program D1, and generates the activation program file D2. In addition, in the activation program file D2, the debug information D3 which shows correspondence with a source program D1 is included. And a parallel processing system 4 operates based on the executive program described in the activation program file D2.
[0040] Moreover, based on the program execution information which shows the program execution situation acquired from a source program D1, the activation program file D2, and a parallel processing system 4, a symbolic debugger 1 grasps the activation situation of a parallel processing system 4 correctly, and displays the debugging display information which is not illustrated by drawing 1 . By this debug information, the debugging activity of a source program D1 becomes easy.
[0041] Moreover, based on the activation trace file D4, instruction combination statistical analysis equipment 3 takes statistics of instruction combination, and outputs the result as a statistics result file D5. Based on a statistics result, a parallel processing system 4 can carry out the combination instruction with high operating frequency to one instruction, and can aim at improvement in a throughput. In addition, instruction combination statistical analysis equipment 3 may also incorporate the activation program file D2 instead of the activation trace file D4.
[0042] < <1st example (symbolic debugger) >> drawing 2 is the block diagram showing the configuration of the symbolic debugger which is the 1st example of this invention. As shown in this drawing, the debug information association table creation means 12 receives a source program D1 and the activation program file D2 (debug information D3 to include), and creates the debug information association table group D10 which expressed information required for debugging in the table format based on these data D1 and D2. In addition, debug information D3 is information including the information about the address of the executive program corresponding to the executable statement in a source program, and the function in a program, the information about the variable in a program, etc.
[0043] If the instruction corresponding to the renewal of a variable on a source program D1 is executed with a parallel processing system 4, the renewal means 13 of the contents of a debug information association table will receive program execution information including the renewal execution information of a variable from a parallel processing system 4, and will update the

contents of the debug information association table group D10 based on the renewal execution information of a variable within program execution information.
[0044] the debug information display means 14 -- the debug information association table group D10 -- being based -- visual recognition -- easy debugging display information is displayed.
[0045] Drawing 3 is a flow chart which shows actuation of the symbolic debugger of the 1st example. With reference to this drawing, the command input to a symbolic debugger 1 is made at step S1. And when an input command is a quit command, processing is ended through branching processing of step S2, and when an input command is an activation initiation command, it moves to the executive operation of symbolic debugger 1 original after step S11 through branching processing of step S3.
[0046] At step S11, the debug information association table creation means 12 performs initial setting by creating the debug information association table group D10 based on a source program D1 and the activation program file D2.
[0047] Moreover, when an input command is an activation continuation command, initial setting of step S11 shifts to processing of step S12 through branching processing of step S4, without carrying out.
[0048] And a parallel processing system 4 is made to perform program actuation based on an executive program at step S12, and it waits for the notice of the program execution information from a parallel processing system 4 at step S12. And in step S14, when it is judged and included whether the information which directs activation termination is included in the program execution information received at step S13, it shifts to step S18, and when not contained, it shifts to step S15.
[0049] The renewal execution information of a variable which shows the contents of activation of the instruction corresponding to the renewal of a variable on a source file is included in program execution information, and a color / generation information is included by this renewal execution information of a variable in the value of the variable after updating, and the address on the executive program of the instruction corresponding to renewal of a variable (there being also a case of a destination node number in the system of a program counter or a data flow method) and the system of a dynamic data flow method.
[0050] In addition, when multiplexing and using the same function as color information, it is an identifier for identifying share function call origin, and when performing based on data which are different in the same program as generation information, it is the information for identifying each.

[0051] The renewal means 13 of the contents of a debug information association table searches the debug information table D11 in the debug information association table group D10 with step S15 based on the notified program execution information, and it checks the renewal [ which source file / which variable ] of the how many lines the renewal of a variable is.
[0052] And the renewal means 13 of the contents of a debug information association table performs updating actuation of the contents of the debug information association table group D10 at step S16.
[0053] Drawing 4 - drawing 7 are the explanatory views showing the detail of the debug information association table group D10. the debug information association table group D10 -- mainly -- the debug information table D11, the variable table D12, a function table D13, and a function -- another line -- it consists of an information table D14. in addition, drawing 4 -- the debug information table D11 -- being shown -- drawing 5 -- the variable table D12 -- being shown -- drawing 6 -- a function table D13 -- being shown -- drawing 7 -- a function -- another line -- the information table D14 is shown. Moreover, the above-mentioned debug information association table group D10 supports source program D1A and D1B which were shown by drawing 8 and drawing 9 .
[0054] the "class" which is the item of the debug information table D11 in drawing 4 -- a function -- another line -- or [ incrementing the information which shows whether it is (C) which increments the count counter of activation of the information table D14, and the loop count to a loop head line ] -- (L) -- the information on how is stored. For example, when one line of a source program D1 has two or more renewal of a variable, the count of activation of each line

can be correctly managed by setting to C only the part corresponding to the renewal of a variable described by the leftmost.

[0055] For example, in the renewal execution information of a variable of program execution information, when the real address directs a number 18, a class is recognized that #2 and a line number is [ the pointer to 5 and a variable table ] #9 for the pointer to C and a function table on the debug information table D11 shown by drawing 4 . And the variable j of func1 in pointer #9 is changed on the variable table D12 shown by drawing 5 (the example of drawing 5 0->1). similarly, as shown in drawing 6 , the renewal line of the newest of the function name func1 of pointer #2 is changed into the 5th line from the 7th line from a function table D13, or it is shown in drawing 7 -- as -- a function -- another line -- the count of activation of the information table D14 is incremented.

[0056] The actuation performed to below based on the entry (a pointer, a line number, etc. are called "applicable entry" below) obtained by retrieval of the debug information table D11 is enumerated.

[0057] - a function -- another line -- increment the count counter of activation of the information table D14.

[0058] - Store the newest value in the variable table D12.

[0059] - Build a flag during activation of a function table D13.

[0060] - The flag with renewal of information of a function table D13 is set to "YES", and it is shown that it is necessary to change at the time of window regeneration.

[0061] - the case of the system of a dynamic data flow method -- a function -- another line -- register the color / generation information on the information table D14 as an executed color / generation information.

[0062] - loop-formation initiation registration and termination registration -- a function -- another line -- carry out to registering with S and F, respectively into the loop-formation reality line finishing ITARESHON list of information tables D14. in the example of drawing 7 , it is nesting 1, the 1st loop formation is completed (1 (1F)), and the loop formation which is the 2nd time starts the attribute loop-formation tail line of #8 (1 (2S)) -- it is ******(ing).

[0063] Moreover, in the renewal of a variable corresponding to loop-formation termination (the case of the loop-formation termination by break, continue, and return is included), the number of nesting (0) is listed from a loop head line to a loop-formation tail line, and it memorizes that the loop formation concerned is in a completion condition.

[0064] It returns to drawing 3 and it is confirmed whether the conditions of an activation break are satisfied at step S17. In addition, the conditions according to which an activation break is materialized point out the following cases.

[0065] - Activation of the count of assignment of stepwise execution was completed. When there is no count assignment, suppose that there was one assignment.

[0066] - The break conditions specified by a user were satisfied.

[0067] - The system error occurred. (The case where actuation is compulsorily stopped by Cntl-C etc. is included.)

When the conditions of a step S17 activation break are not satisfied, it waits for the next notice of renewal of a variable at step S18. On the other hand, when the conditions of an activation break are satisfied, it shifts to step S18.

[0068] The debug information display means 14 expresses the function that whose the flag serves as YES during activation of a function table D13 it was in the condition of not displaying as step S18. However, when there is an attention assignment function, the new display of those other than an attention assignment function is iconified. (However, an icon display is not performed when the flag which is not iconified has left.)

Moreover, when the debug information display means 14 displays each function, the display part of a function displays that the line performed immediately before is included. When there is a renewal part of each line or the variable of plurality [ line / one ], the field is expressed as the color corresponding to the count counter of activation. Furthermore, it emphasizes by performing an arrow-head display in the line which had renewal of a variable immediately before. When the function of relevance is an icon display, only the count of renewal of a variable is made to blink.

[0069] Moreover, the debug information display means 14 also performs the display of loop count in a loop count display column. In using a dynamic data flow processor as a target machine, a color / generation information also performs rewriting and a display of an executed color / generation information table.
[0070] Drawing 10 is the example of a display of the debug information display means 14. As shown in this drawing, it expresses as different colors C1-C3 ( drawing 10 hatching) for every count of activation, an arrow-head display is performed in the line which had renewal of a variable immediately before, and a generation's display (G4, G5 grade) and the display of loop count are performed.
[0071] moreover, the identifier of recursive function instruction statement [ in / in drawing / based on color information / for the nesting situation in the high recursive function of possibility that the debug information display means 14 will be frequently called as debugging display information / display sushi and a source program ] -- recognition -- it can also be made easy.
[0072] After completing the activity of a display of step S18, again, it becomes command processing of step S1 with return, and becomes a command with an input state. In command processing of step S1, the processing perform commands, such as modification of the value of a variable, like the symbolic debugger of the command former which directs modification of the noicon command which inhibits the iconification of an attention function directive command and a non-observing function other than the quit command and the activation initiation command mentioned above in addition to the usual debugger, the command which performs the detail display of loop-formation activation and the detail display of a color / generation activation, and the attention assignment function of an activation situation is included.
[0073] executed in un-performing, although the 1st example of which the <modification> above was done explained the case where the debug information display means 14 changed a display for every count of activation -- those two kinds -- you may make it display When the processing which carries out back track on a source program by the loop formation or jump instruction in performing two kinds of discernment occurs, about the part which retraced its steps, an executed part is also changed and displayed on non-performed sector display.
[0074] Moreover, although the 1st example showed the example into which a foreground color is changed for every count of activation, whenever it performs once, it may be made to indicate to the left of the line of each source program by the count. Furthermore, you may display that one character is added to the left of the line of each source program whenever it performs once.
[0075] Moreover, as are shown in drawing 11 , and the debug information display means 14 displays the notice situation of activation in front of each function as a histogram, it may constitute it so that the function under present activation can be recognized with the operating frequency. Moreover, based on the just before use frequency, the function with which the count of predetermined was recorded may be constituted so that it may be shown a blink table.
[0076] Consequently, an activation situation more exact than the function under current activation can be recognized by the comparison of the count of activation just before each of each function can be grasped to a parallel processing system.
[0077] The sequence described on the source file the program which operates on parallel processing systems, such as a <effectiveness> data flow processor, a processor of a super scalar method or a multithread method, and a multiprocessor system, was described to be by the program differs from the sequence actually performed. Moreover, two or more parts on a source file are performed by coincidence. Moreover, activation of the object optimized even if it was a debugger on the processor of the usual J. von Neumann form serves as complicated sequence.
[0078] On the other hand, the debug information display means 14 of the symbolic debugger of the 1st example can perform discernment of the part (instruction statement) which is not performed [ an executed line (instruction statement) and ] on a source program, and displays the debugging display information in which the discernment which is a count of activation is possible. Furthermore, identifiable debugging display information can also be displayed [ a color / generation information ] for loop count and the count of activation just before a function unit to the part (instruction statement) by which multiple-times activation may be carried out.
[0079] Consequently, the symbolic debugger of the 1st example Since discernment from

executed instruction statement and non-performed instruction statement, the count of activation of each instruction statement, a color / generation information, loop count, and the count of activation just before a function unit can be recognized by referring to debugging display information, The exact display of complicated processing of the above-mentioned parallel processing system is realized, and the effectiveness that the environment which can be debugged can be offered like the system performed as the description on a source file is done so.

[0080] < <2nd example (compiler 2) >> drawing 12 is the block diagram showing the internal configuration of a compiler 2. As shown in this drawing, the source program reading means 21 reads a source program D1, and outputs it to the synchronizing primitive generation means 22 classified by condition.

[0081] The synchronizing primitive generation means 22 classified by condition generates the program code group which becomes the origin of an executive program based on a source program D1. Under the present circumstances, even if it is the same synchronizing primitive, when conditions differ, a program code which is different as two or more sorts of synchronizing primitive codes according to a condition is generated.

[0082] Based on a program code group, taking the classification of a synchronizing primitive into consideration, the optimization processing means 23 performs optimization processing, and outputs an optimized program code group to the synchronizing primitive conversion means 24.

[0083] To an optimized program code group, the synchronizing primitive conversion means 24 changes two or more sorts of synchronizing primitives into a common synchronizing primitive code, and outputs an executive program to the activation program file D2.

[0084] Drawing 13 is a flow chart which shows actuation of the compiler 2 which is the 2nd example. First, at step S21, the source program reading means 21 reads a source program D1, and outputs it to the synchronizing primitive generation means 22 classified by condition.

[0085] And the synchronizing primitive generation means 22 classified by condition generates the program code group which becomes the origin of an executive program at step S22 based on a source program D1. Under the present circumstances, at the time of the code generation of synchronous processing ******, the synchronizing primitive code of a different classification for every conditions is generated.

[0086] Next, optimization processing is performed to a program code group at step S23 after completing all the code generation generated at step S22. In optimization processing, processing which deletes all of whether unnecessary synchronous processing is contained because the synchronous processing for every conditions laps, and the processing which checks and deletes unnecessary synchronous processing and the synchronous processing of those other than a critical path is performed. Moreover, to the execution file after a link, the synchronous processing before and behind a function call judges truly whether it is the need, and when unnecessary, the processing to delete is included.

[0087] Hereafter, processing of step S23 is explained in full detail. As a synchronizing primitive according to condition, there are synchronizing primitive sync-tsk which determines the activation schedule for every processing, and synchronizing primitive sync-dev which uses a certain device exclusively. As shown in drawing 14 , after a task T1 and a task T2 synchronize by synchronizing primitive sync-tsk, it moves from them to processing of task T3.

[0088] However, as shown in drawing 15 , while moving to processing of task T3 after synchronizing by synchronizing primitive sync-tsk, a task T1 and a task T2 When the program code group is generated so that it may move to processing of task T3 after a task T1 synchronizes by synchronizing primitive sync-dev, Since only synchronizing primitive sync-tsk can perform synchronous processing of a task T1 convenient, it can be made an efficient program code group by judging with synchronizing primitive sync-dev being unnecessary, and deleting.

[0089] Moreover, in the instruction which refers to the contents of memory, such as a substitution instruction of a variable, it is necessary to clarify the context according to the contents. For example, when the substitution instruction a=b;c=a; continues, it is necessary to surely perform "a=b" ahead of "c=a." Thus, the synchronizing primitive for maintaining the

sequentiality of memory access is called synchronizing primitive sync-mem.
[0090] Therefore, when a reference instruction occurs, as shown in drawing 16 , a program code group is generated by the synchronizing primitive generation means 22 classified by condition so that synchronizing primitive sync-mem may perform synchronous processing of a reference instruction R1 and a reference instruction R2 and synchronizing primitive sync-mem may perform synchronous processing with the processing after the synchronization, and a reference instruction R3 further. in addition, a reference instruction is an instruction which obtains with "c=a", is and calls the contents of the variable a in a substitution instruction.
[0091] However, a program code group is analyzed, and when it has been recognized that the same processor surely performs in order of reference instructions R1 and R2, reference instructions R1 and R2 will end, if synchronizing primitive sync-mem performs synchronous processing with a reference instruction R2 and a reference instruction R3 as shown in drawing 17 . that is, the part which can omit synchronizing primitive sync-mem of a reference instruction R1 and a reference instruction R2 -- it optimizes.
[0092] Moreover, in the code which makes coincidence juxtaposition perform the same function using an identifier color (share function), before releasing the color gained in advance of function activation, all function outputs are outputted, and in order to check that the color has become no more use, it is necessary to take a synchronization. Synchronizing primitive sync-freec is outputted at this time.
[0093] Optimization made into an efficient program code group can be performed by changing into the processing which gives a color by carrying out in RIMENTO of the value of a predetermined register for the predetermined instruction-execution processing which the execution time requires to the case where the number of need colors is limited to the call code of this share function and which is taken out color picking. Such color synchronous optimization processing is indicated by for example, the Japanese-Patent-Application-No. No. 155182 [ five to ] official report.
[0094] If optimization processing color synchronous [ above-mentioned ] is performed, all synchronizing primitive sync-freec can be deleted. That is, by specifying and coding synchronizing primitive sync-freec, synchronizing primitive sync-freec can be deleted easily and optimization can be attained.
[0095] After all optimization processing including the above-mentioned example is completed, at step S24, the synchronizing primitive conversion means 24 changes into a common synchronizing primitive code all the synchronizing primitive codes set as a different classification, and outputs them as an executive program.
[0096] <Effectiveness> In this way the compiler 2 of the 2nd example On the occasion of optimization processing, change a synchronizing primitive code according to a condition, and the program code group which becomes the origin of an executive program is generated. Since the optimization processing of an executive program becomes easy and the procedure of unnecessary synchronous processing or a function call can be deleted by performing optimization processing based on this program code group, and transposing the synchronizing primitive classified after that to a common code It becomes compressible [ improvement in the speed of the processing in a parallel processing system, and the activation program file D2 ].
[0097] Especially, in code of a data flow processor, although the synchronous code was inserted in the loop code generate time for every ITARESHON, it leaves an indispensable synchronous code and it also becomes possible to achieve optimization of a code.
[0098] Although code classification according to condition of a synchronizing primitive was performed and optimization was attained in the 3rd example of a <modification>, code classification according to call condition of a function can be performed, and optimization can be attained. For example, it optimizes so that it may be made to jump to the function of the direct bottom, when the code of these functions is separated for call conditions when it calls once and a function is classified according to the activation program size minimum etc., a nesting condition and, a program code group is generated and a function call carries out nesting, and when an activation program size is very small, and a function call is 1 time, inline expansion can be performed and optimization can be attained.

[0099] < <3rd example (instruction combination statistical analysis equipment 3) >> drawing 18 is the block diagram showing the configuration of instruction combination statistical analysis equipment 3. As shown in this drawing, the activation trace file reading means 31 reads the activation trace file D4, and outputs it to the run command data generation means 32.
[0100] Based on the activation trace file D4, the run command data generation means 32 creates a run command data table, and outputs it to the data dependency recognition means 33.
[0101] The data dependency recognition means 33 recognizes the data dependency between run commands with reference to a run command data table, writes a data dependency in a run command data table, and outputs a run command data table with data dependency information to the statistical analysis means 34 classified by check pattern.
[0102] On the other hand, the mode assignment means 30 determines the mode of statistical analysis based on the command from the outside, and outputs it to the statistical analysis means 34 classified by check pattern by making the mode into assignment mode (statistical analysis command information).
[0103] The statistical analysis means 34 classified by check pattern performs statistical analysis processing with the assignment mode obtained from the mode assignment means 30 based on a run command data table with data dependency information, and outputs the statistics result to the statistics result output means 35.
[0104] The statistics result output means 35 outputs the statistics result file D5 based on the statistics result obtained from the statistical analysis means 34 classified by check pattern.
[0105] Drawing 19 is a flow chart which shows activation actuation of the instruction combination statistical analysis equipment 3 which is the 3rd example. With reference to this drawing, the mode assignment means 30 performs initial setting according to reading and assignment mode of the command line of a command which were obtained from the exterior at step S31.
[0106] The file designation for statistics and check pattern assignment of about what kind of pattern to take statistics are included in a command line.
[0107] In the file designation for statistics, multiple-files assignment is possible and the file type in which file designation is possible is two kinds. In two kinds, there are an object program file and an activation trace file which filed the activation hysteresis of a parallel processing system 4. Two kinds of files may be intermingled at the time of multiple-files assignment. Whether it is an object program file or it is an activation trace file recognize by the magic number or file classification number in the extension (an object and .trc trace for example, .o) of a file, or a file header. According to each file format, it reads according to recognition.
[0108] One kind of assignment is possible for check pattern assignment among three kinds of assignment. The 1st kind of check pattern assignment is the minimum assignment of the number of combination instructions, and the maximum assignment (mode 1). The 2nd kind of check pattern assignment is assignment for checking to perfect pattern assignment, a specific instruction, or the combination pattern of an instruction group (mode 2). The 3rd kind of assignment is assignment of the maximum of the sum total of the execution time according to the number according to class of instruction, or class of instruction (mode 3).
[0109] "sinst [[[-D|P], [-cmax #], and [-cmin #]] |[-p file.pat] |[-s file.cmb]] files"
**** is a format of a command line. When [] shows an omissible part and it is divided by |, if it is "-D|-P" and any of -D or -P are "A|B|C", it is shown that any of A, B, and C are specified. sinst is an invocation command and files is a file designation part for statistics. Except [ its ] is check pattern assignment. Although check pattern assignment is omissible, when all are omitted, it serves as the same actuation as assignment of "-D-cmax 3-cmin 2." This is assignment in the mode 1 and means that statistics is taken about the pattern of the instruction combination which has a data dependency between instructions by three or less instruction combination or more 2 number.
[0110] Next, it explains according to each class of each check pattern assignment.
[0111] A format of the 1st (mode 1) kind of check pattern assignment is "[-D|-P], [-cmax #], and [-cmin #]."
It comes out. - When -P is specified about the instruction combination which has a data

dependency when D is specified, take statistics about the instruction combination which can be performed to coincidence juxtaposition. - When assignment of D does not have assignment of -P, either, process as what -D was specified as. It is specified as the figure to which the maximum number of the combination number of the instruction which takes statistics follows -cmax, and the minimum number of the combination number of the instruction which takes statistics is specified as the figure following -cmin. When these number assignment is omitted, it becomes the same actuation as the case where assignment of -cmax 3-cmin 2 is made. When the 1st kind of this check pattern assignment is performed, a mode of operation is set as 1.

[0112] A format of the 2nd (mode 2) kind of check pattern assignment is "-p file.pat."
It comes out. - If p is followed, a pattern designated file name is specified. In the pattern designated file, the combination and the data dependency of the instruction and instruction group which are checked as check pattern, or an instruction class are described. In this file, it is possible to describe two or more check pattern. When this pattern designated file is specified, based on description, a check pattern table is created within step S31. When the 2nd kind of this check pattern assignment is performed, a mode of operation is set as 2.

[0113] Next, the example of a pattern designated file is shown.

[0114] "-- PAT1; -- pattern 1INC VAL1MUL VAL2 VAL1-AT2; -- pattern 2DIV * *ALU * *ALU * *-AT3; -- pattern 3 SUB|ADD VAL1 VAL2MUL VAL3 VAL1END"
Three check pattern is described in this file. First, a pattern 1 is description of the pattern which has a dependency to which MUL (multiply operation) of the result of INC (increment instruction) is carried out. This will be counted if there are INC and MUL with the data dependency described when there was a different instruction execution not only between when INC and MUL are performed continuously, but activation of INC and MUL.

[0115] A pattern 2 is a pattern in which DIV (divide operation) and two ALU run commands which can be executed are shown mutually-independent. (It is shown that * of an operand does not have a data dependency.) It is here and an instruction class is indicated to be ALU.

[0116] A pattern 3 is a pattern in which MUL which has a data dependency in SUB (subtraction instruction) or ADD (add instruction), and it is shown. Here, although two instruction groups, SUB and ADD, are specified, three or more instruction groups can also be specified.

[0117] A format of the 3rd (mode 3) kind of check pattern assignment is "-s file.cmb."
It comes out. - If s is followed, a combination designated file name is specified. The limits (the data-access instruction, the FALU instruction, control instruction, etc.) of activation total time amount or the number according to instruction class is described in the combination designated file. In carrying out limit assignment of activation total time amount, when you want to take statistical information for the execution time of each instruction except a default, it is necessary to put instruction-execution-time designated file"INST.DATA" which described the execution time of each instruction on an activation directory. If there is "no INST.DATA", it will process in the default execution time. A check pattern table is generated according to a combination designated file. When the 3rd kind of this check pattern assignment is performed, a mode of operation is set as 3. When a mode of operation is 3, if instruction-execution-time designated file"INST.DATA" is looked for and it is all over an activation directory, the execution time of each instruction will be read.

[0118] After initial setting, such as processing of the command line of step S31 and mode setting, is completed, the activation trace file D4 which is a file for statistics is read at step S32, and every one read run command is stored in a run command data table.

[0119] Next, at step S33, the contents of the run command data table are analyzed and data dependency information is stored in a run command data table. in addition, when the file for statistics does not come out activation trace file D4 and there are read-out from indefinite address memory and writing by the activation program file D2, it is judged, assuming decision whether parallel execution is possible that there were read-out in the same memory and writing. About a data dependency, it judges as R/W to different memory.

[0120] Drawing 20 is the explanatory view showing some run command data tables. As shown in this drawing, an entry number, a run command, etc. are described by the entry column, and the entry number which has a data dependency at the data dependency column is described. In the

example of drawing 20 , since a data dependency is between the ADD instruction of the entry number 1, and the SUB instruction of the entry number 2 about "r5", #2 are written in the data dependency column of an ADD instruction. After the above processing finishes, it is confirmed at step S34 and step S35 whether it is in agreement with the pattern of a check pattern table about all the instruction combination that contains each entry from the 1st entry of a run command data table.

[0121] In the mode 2 and the mode 3, all the entries of a check pattern table are checked for every entry. However, in the case of the mode 1, it checks sequentially from the variation of the minimum combination of the number of instruction combination. the check of all instruction combination (in the case of the mode 1), or all entries -- completing (when it being the modes 2 and 3) -- processing shifts to step S36.

[0122] Hereafter, the detail of processing of step S35 is explained. At step S35, processing is performed according to the mode.

[0123] In the case of the mode 1, based on a check pattern table, it checks to each instruction combination of a run command data table, and in being registered, the counter of an applicable pattern is incremented, and in not registering, additional registration is carried out and it sets the counter of an applicable pattern as a check pattern table 1. The identifier of the check pattern which incremented the counter is stored in the run command data table corresponding to each instruction of the instruction combination used as the candidate for a current check at this time. By this, the same instruction avoids counting 2 times or more with the same check pattern.

[0124] That is, when instruction sequence called ADD, SUB, ADD, SUB, and ADD exists in a run command data table, the number of instruction combination has taken with statistics to 3, and ADD, SUB, and ADD are checked first, to ADD, SUB, and ADD, it becomes finishing counting 3rd ADD and it is counted once by instruction sequence called ADD, SUB, ADD, SUB, and ADD.

[0125] In the case of the mode 2, if all the entries of a check pattern table are checked and there is a match from the 1st entry of a run command data table, the count of the pattern will be incremented one time. In the case of check pattern including an instruction group or an instruction class, a subcheck pattern table is created, and the count to each instruction is also carried out to it.

[0126] In the case of the mode 3, if all the entries of a check pattern table are checked and there is a match from the 1st entry of a run command data table, the count of the pattern will be incremented one time. It will restrict, when in agreement, and in order to take statistics about the instruction combination which was actually in agreement with the subcheck pattern table for every pattern, if the subcheck pattern table for every pattern is also checked and there is, a counter will be incremented, if there is nothing, it will newly register and a counter will be set as 1. In step S36, it moves to processing of the following file for statistics. After processing of all the files for statistics is completed, at step S38, statistics processing for every file and the whole statistics processing are performed, and a result is outputted.

[0127] In processing to two or more files for statistics, whenever it completes processing of one file, after copying the counter of a check pattern table and a subcheck pattern table to a backup area, in step S37, it initializes to zero. And it repeats from processing of step S32 next.

[0128] The statistical analysis means 34 classified by check pattern of the instruction combination statistical analysis equipment 3 of the 3rd example of <effectiveness> The run command data table with a data dependency and assignment mode which were created based on the activation trace file D4 are received. Since the count of activation of the instruction combination which assignment mode directs with reference to a run command data table with a data dependency is counted, statistical analysis is performed and a statistical analysis result is outputted, the activation frequency of the instruction combination concerned can be recognized by referring to a statistical analysis result.

[0129] Consequently, it can grasp comparatively easily which instruction combination should be made one instruction by setting up so that statistical analysis command information may direct the instruction combination which asks for statistical analysis, and making a statistical analysis result output with instruction combination statistical analysis equipment 3.

[0130] If the instruction combination which can be performed to juxtaposition as combination

which asks for statistical analysis especially is directed, the activation frequency of the instruction combination which can be performed to juxtaposition can be recognized by referring to a statistical analysis result.
[0131] Moreover, if the instruction combination which has a data dependency as combination which asks for statistical analysis is directed, the activation frequency of instruction combination with a data dependency can be recognized by referring to a statistical analysis result.
[0132] Thus, in case the processor for performing specific application by taking various statistical analysis with instruction combination statistical analysis equipment 3 is developed, what kind of compound instruction is effective? It can opt for the optimal numbers, such as ALU, FALU, etc. which are arranged to an execution unit, and the optimal connection.
[0133] Moreover, it can judge whether a VLIW (Very Long instruction word) method is advantageous or it is advantageous that there is a compound instruction by the multistage pipelined architecture.
[0134] Moreover, since statistical analysis can also be performed based on the activation program file D2, instruction combination statistical analysis equipment 3 is this time, and even when using as a target a system which has the application which can take trace data, and the application which cannot be taken, although it doubled both, it can obtain a statistical data.
[0135] When advantageous architecture is determined from the statistical analysis result by the instruction combination statistical analysis equipment 3 of the 3rd example of a <modification> Carry out taking statistics, although the execution time of the instruction combination in the architecture is settled within assignment time amount etc., consider the prediction function of the execution time, and it is made to output the statistics result file D5. You may make it the configuration which can also acquire the information about the execution time of the instruction combination concerned by referring to the statistics result file D5.
[0136] When the application frequently performed when the executive program consists of two or more partial programs (application), and the application seldom performed are contained, activation frequency data may also be considered, and although the 3rd example explained the case where totaled simply the data taken from each file, and it considered as whole statistics, you may constitute so that a whole statistics result may be outputted. In this case, it can grasp which instruction combination it is adapted for every application and should be made one instruction.
[0137]
[Effect of the Invention] In debugging equipment [ in / as explained above / this invention ] according to claim 1 computing system performance information The information which identifies the existing run statement to which activation was already performed among two or more instruction statement of a source program, and the non-run statement which is not yet performed is included. Since computing system display information is displayed that the existing run statement and the non-run statement in a source program can identify a display means visually, the existing run statement and the non-run statement in a source program can be easily recognized only by referring to computing system display information.
[0138] Consequently, even if a computing system is a parallel processing system, a more exact activation situation can be grasped.
[0139] It sets to the debugging equipment according to claim 2 in this invention, and since the count of activation displays that computing system display information is visually discriminable to two or more instruction statement of each which can set a display means to a source program including the information from which computing system performance information discriminates the count of activation of two or more instruction statement of each of a source program, the count of two or more instruction statement of each, which can be easily set to a source program only by referring to computing system display information, of activation can recognize.
[0140] Consequently, even if a computing system is a parallel processing system, a more exact activation situation can be grasped.
[0141] In the debugging equipment according to claim 3 in this invention computing system performance information A display means receives each instruction statement within a loop formation including the information which shows the activation hysteresis of each instruction

statement within a loop formation which is the instruction statement which forms a loop formation among two or more instruction statement. Since the activation hysteresis displays that computing system display information is visually discriminable, activation situations, such as loop count of the instruction statement within a loop formation in a source program, can be easily recognized only by referring to computing system display information.
[0142] Consequently, even if a computing system is a parallel processing system, a more exact activation situation can be grasped.
[0143] In the debugging equipment according to claim 4 in this invention, since computing system display information is displayed that an identifier can identify a display means visually corresponding to share function instruction statement including the information about an identifier for computing system performance information to identify share function call origin, the identifier of the share function instruction statement in a source program can be easily recognized only by referring to computing system display information.
[0144] Consequently, even if a computing system is a parallel processing system, a more exact activation situation can be grasped.
[0145] Moreover, since the above-mentioned share function displays computing system display information that the account identifier of the high recursive function of possibility of being called frequently can identify visually, debugging equipment according to claim 5 can recognize the identifier of the recursive function instruction statement in a source program easily only by referring to computing system display information.
[0146] In the debugging equipment according to claim 6 in this invention computing system performance information In order to display computing system display information that the count of activation just before two or more functions can identify a display means visually including the information which identifies the count of activation just before being a count of activation in a past fixed period of a just before [ two or more functions ], The count of activation just before two or more functions which can be easily set to a source program only by referring to computing system display information can be recognized.
[0147] Consequently, even if a computing system is a parallel processing system, an activation situation more exact than the function under current activation can be recognized by the comparison of the count of activation just before two or more share functions of each can be grasped.
[0148] Since the program code group generation means of the compiler according to claim 7 concerning this invention generates a different program code for every generation condition of that to the specific instruction statement in two or more instruction statement, an optimization processing means can perform the part which can perform optimization processing in consideration of the generation condition of specific instruction statement, and more advanced optimization processing.
[0149] Consequently, since an optimization processing means can perform advanced optimization processing, finally an efficient executive program is generable.
[0150] Since especially a compiler according to claim 8 is the above-mentioned specific instruction synchronizing primitive, it can generate an efficient executive program for a predetermined computing system to perform a parallel processing system.
[0151] The statistical-analysis means in the statistical-analysis equipment according to claim 9 concerning this invention receives the information for statistics, and statistical-analysis command information, and since it counts the count of activation of the instruction combination which statistical-analysis command information directs with reference to the information for statistics, performs statistical analysis and outputs a statistical-analysis result, it can recognize the activation frequency of the instruction combination concerned by referring to a statistical-analysis result.
[0152] Consequently, it can grasp comparatively easily which instruction combination should be made one instruction by setting up so that statistical analysis command information may direct the instruction combination which asks for statistical analysis, and making a statistical analysis result output.
[0153] Moreover, since the statistical analysis command information on statistical analysis

equipment according to claim 10 is information which directs the instruction combination which can be performed to juxtaposition, it can recognize the activation frequency of the instruction combination which can be performed to juxtaposition by referring to a statistical analysis result.
[0154] Moreover, since the statistical analysis command information on statistical analysis equipment according to claim 11 is information which directs instruction combination with a data dependency, it can recognize the activation frequency of instruction combination with a data dependency by referring to a statistical analysis result.
[0155] Moreover, since the statistical analysis means of statistical analysis equipment according to claim 12 also performs statistical analysis about the execution time of the instruction combination which has a data dependency further, it can also acquire the information about the execution time of the instruction combination concerned by referring to a statistical analysis result.
[0156] Moreover, since a statistical analysis means according to claim 13 outputs the statistical analysis result in two or more partial-program units, it can grasp which instruction combination it is adapted for every partial program, and should be made one instruction.

---

* NOTICES *

JPO and NCIPI are not responsible for any
damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]
[Drawing 1] It is the mimetic diagram showing the whole development support system configuration of a parallel processing system.
[Drawing 2] It is the block diagram showing the configuration of the symbolic debugger which is the 1st example of this invention.
[Drawing 3] It is the flow chart which shows actuation of the symbolic debugger of the 1st example.
[Drawing 4] It is the explanatory view showing the debug information table of a debug information association table group.
[Drawing 5] It is the explanatory view showing the variable table of a debug information association table group.
[Drawing 6] It is the explanatory view showing the function table of a debug information association table group.
[Drawing 7] the function of a debug information association table group -- another line -- it is the explanatory view showing an information table.
[Drawing 8] It is the explanatory view showing a part of source program.
[Drawing 9] It is the explanatory view showing a part of source program.
[Drawing 10] It is the explanatory view showing an example of debugging display information.
[Drawing 11] It is the explanatory view showing an example of debugging display information.
[Drawing 12] It is the block diagram showing the configuration of the compiler which is the 2nd example of this invention.
[Drawing 13] It is the flow chart which shows actuation of the compiler of the 2nd example.
[Drawing 14] It is an explanatory view for explanation of the compiler of the 2nd example of operation.
[Drawing 15] It is an explanatory view for explanation of the compiler of the 2nd example of operation.
[Drawing 16] It is an explanatory view for explanation of the compiler of the 2nd example of operation.
[Drawing 17] It is an explanatory view for explanation of the compiler of the 2nd example of operation.
[Drawing 18] It is the block diagram showing the configuration of the instruction combination statistical analysis equipment which is the 3rd example of this invention.
[Drawing 19] It is the flow chart which shows actuation of the instruction combination statistical analysis equipment of the 3rd example.
[Drawing 20] It is the explanatory view showing an example of a run command data table.
[Description of Notations]
11 A data reading means, 12 A debug-information association table creation means, the renewal means of the contents of 13 debug-information association table, 14 A debug-information display means, 21 source-program reading means, 22 The synchronizing primitive generation means classified by condition, 23 An optimization-processing means, 24 A synchronizing primitive conversion means, 31 An activation trace file reading means, 32 A run command data generation

means, 33 A data dependency recognition means, 34 The statistical analysis verification means classified by check pattern, 35 Statistics result output means.

---

[Translation done.]